



Acara 1

Materi Pembelajaran : Instalasi Composer dan GitHub
Acara Praktikum/Praktik : 1/1
Tempat : Daring/Luring Politeknik Negeri Jember
Alokasi Waktu : 2 x 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

- Mahasiswa mampu memahami dan mengimplementasikan instalasi Composer
- Mahasiswa mampu memahami dan mengimplementasikan GitHub sebagai repositori
- Mahasiswa mampu memahami dan mengimplementasikan aplikasi version control Git

b. Penilaian Bertahap BNSP (*Skill Passport*)

Nama Skema Sertifikasi: Pengembang Web/Web Developer (SKM-496-028)

No	Kode Unit	Nama Unit Kompetensi	Elemen Kompetensi
1	J.620100.001.01	Menganalisis Tools	<ul style="list-style-type: none">• Mengidentifikasi tools yang akan digunakan• Menggunakan tools perangkat lunak
2	J.620100.002.01	Menganalisis Skalabilitas Perangkat Lunak	<ul style="list-style-type: none">• Menggunakan tools perangkat lunak• Menganalisis kebutuhan skalabilitas

c. Indikator Penilaian

- Ketepatan dalam memahami dan mengimplementasikan instalasi Composer
- Ketepatan dalam memahami dan mengimplementasikan GitHub sebagai repositori
- Ketepatan dalam memahami dan mengimplementasikan aplikasi version control Git

d. Dasar Teori

Composer

Composer berfungsi untuk menghubungkan project aplikasi dengan library dari packagist.org. Packagist merupakan situs yang menyimpan banyak library PHP yang bisa digunakan melalui Composer.

Selain menghubungkan dengan Packagist, Composer punya banyak manfaat untuk Anda. Beberapa di antaranya adalah:



- Membuat programming dengan PHP jadi lebih terstruktur dengan konsep MVC.
- Composer punya fitur autoload sehingga Anda tidak perlu memasukkan semua class PHP yang dibutuhkan.
- Anda tidak perlu memasukkan package secara manual karena semua package sudah tersedia otomatis di Composer.

GitHub

Sebelum kita membuat repository di github lebih baiknya kita mengetahui perintah-perintah dasar github, yaitu :

- git init** : untuk membuat repository lokal di dalam folder project.
- git clone** : meng-clone atau meng-copy projek dari repository
- git status** : untuk menampilkan status pada repository lokal.
- git add** : menambah file baru pada repository yang dipilih.
- git commit** : menyimpan perubahan, dan setiap perubahan ini wajib memberikan keterangan pada setiap perubahan.
- git push** : mengirimkan perubahan file setelah di commit ke repository.
- git pull** : mengambil file yang sudah di ubah dan di upload.
- git branch** : melihat branch yang tersedia pada repository.
- git merge** : menggabungkan semua branch yang ada pada repository.

e. Alat dan Bahan

- Client Server: xampp
- Text Editor: Visual Code/Sublime Text 3
- Management Storage: Git dan Github
- Tools Dependency Manager Multiplatform: Composer
- Kertas A4 / Folio Bergaris
- Pulpen

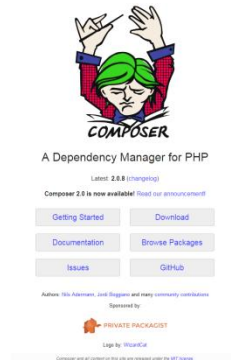
f. Prosedur Kerja

INSTALASI COMPOSER

1. Download File Composer

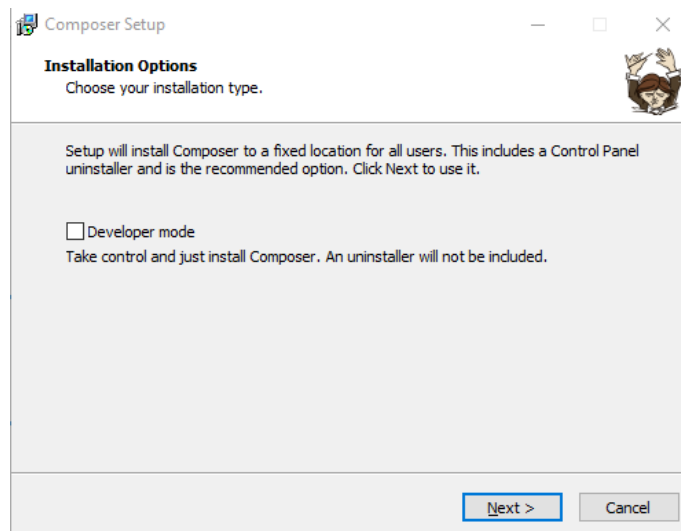


Sebelum melakukan instalasi, hal pertama yang harus dilakukan adalah mengunduh file Composer. Buka link <https://getcomposer.org/> untuk mengunduh file tersebut.



2. Install Composer

Pertama akan muncul halaman seperti pada gambar dibawah ini. Klik **Next** untuk melanjutkan ke proses instalasi.

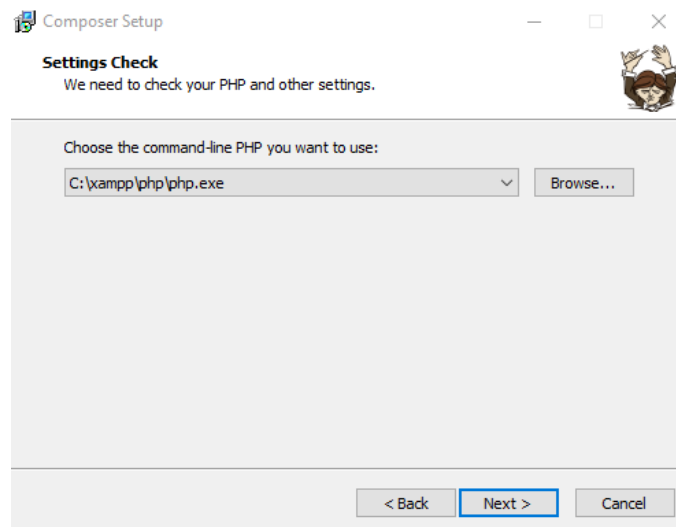


3. Tentukan Lokasi File Instalasi

Selanjutnya Anda akan diminta untuk menentukan lokasi file php (php.exe) yang akan digunakan pada command line (PHP CLI- Command Line Interface).

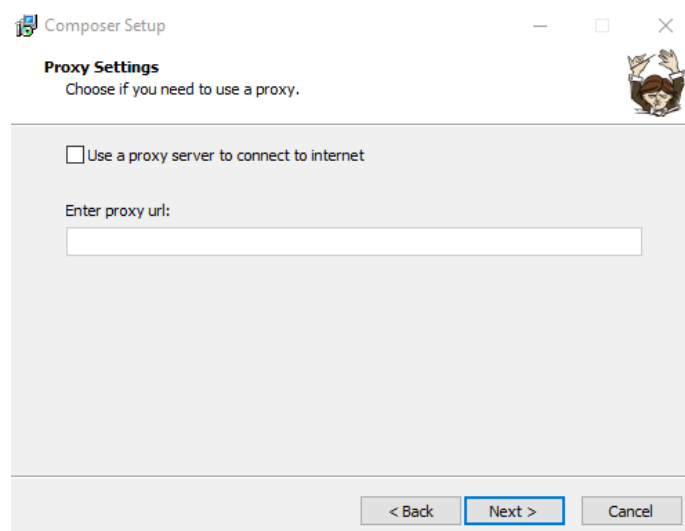


File tersebut biasanya berada di dalam folder php. Jika Anda menggunakan XAMPP, lokasi file ini akan berada di C:\xampp\php. Pastikan pilih lokasi instalasi Composer berjalan di C:\xampp\php\php.exe. Klik Next apabila lokasi file php sudah benar.



4. Pilih Penggunaan Proxy

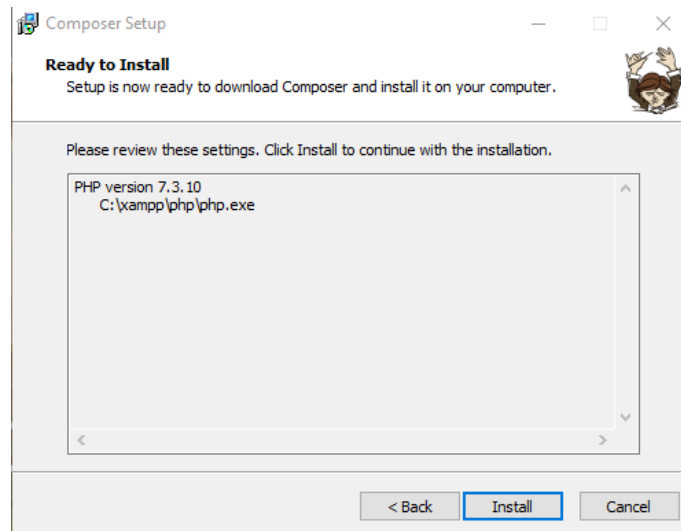
Kemudian Anda akan diminta untuk memilih, apakah Anda akan menggunakan proxy atau tidak. Jika Anda ingin menggunakan proxy, klik centang dan masukkan URL proxy Anda. Apabila Anda tidak ingin menggunakan proxy, langsung klik Next untuk melanjutkan instalasi.



5. Review Lokasi Instalasi

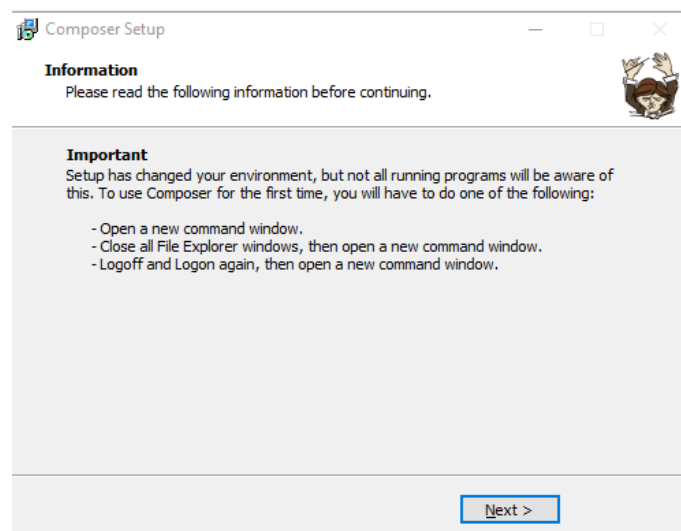


Selanjutnya pastikan kalau proses instalasi berjalan di lokasi file yang seharusnya, yaitu C:\xampp\php\php.exe. Jika sudah benar, klik Install.



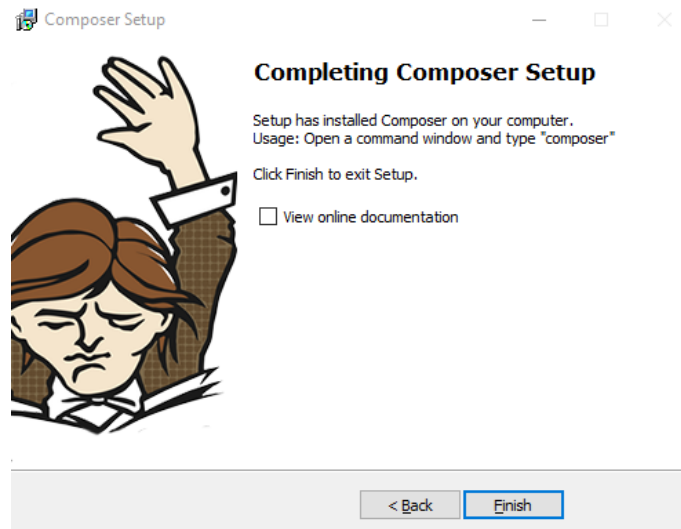
6. Review Perubahan Instalasi

Berikutnya adalah tampilan pemberitahuan bahwa telah terjadi perubahan pada Windows environment. Perubahan ini dimaksudkan agar Composer dapat dijalankan pada Command Prompt.



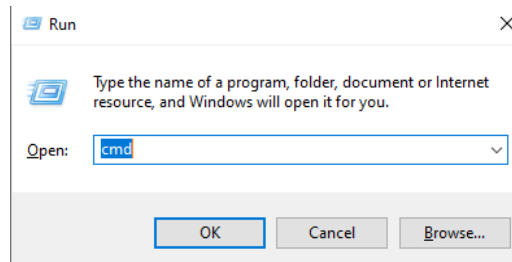
7. Instalasi Selesai

Proses instalasi selesai, klik **Finish** untuk menutup jendela instalasi Composer.



8. Cek Instalasi Composer

Setelah proses instalasi Composer selesai, langkah selanjutnya adalah cek instalasi Composer menggunakan Command Prompt. Caranya adalah Tekan Win+R lalu ketik cmd dan klik Ok.



Maka Anda akan diarahkan ke jendela Command Prompt masukkan perintah seperti di bawah ini untuk mengecek instalasi sukses atau tidak.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\dafit>composer

Composer version 1.9.0 2019-08-02 20:55:32

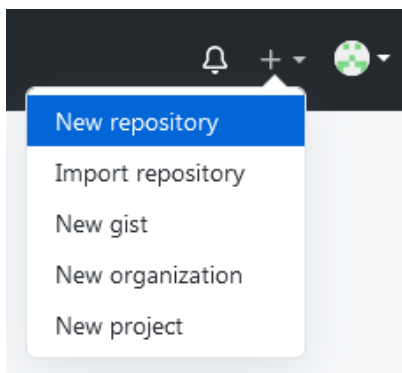
Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                             3 for debug
```

Apabila muncul seperti gambar di atas, instalasi Anda berhasil.

MEMBUAT REPOSITORY DAN UPLOAD PROJECT KE GITHUB

1. Login ke github.com dan buatlah sebuah repository baru dengan mengeklik tombol yang terletak pada kanan atas atau pada sidebar karena pertama kita membuat repository seperti gambar berikut:




Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository

[Import repository](#)



2. Buat repository baru, misalnya dengan nama "produktif" kemudian klik tombol  berikut tampilan Create a new repository :



A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Create a new repository


Owner * Repository name *


 labksi / produktif 

Great repository names are short and memorable. Need inspiration? How about [legendary-guacamole?](#)

Description (optional)

Website Pameran Program Studi Teknik Informatika

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

3. Sekarang kita bisa mengakses remote repository dengan url
<https://github.com/labksi/produktif.git>



Search or jump to... Pull requests Issues Marketplace Explore

labksi / produktif Unwatch Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/labksi/produktif.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# produktif" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/labksi/produktif.git
git push -u origin main
```

...or push an existing repository from the command line

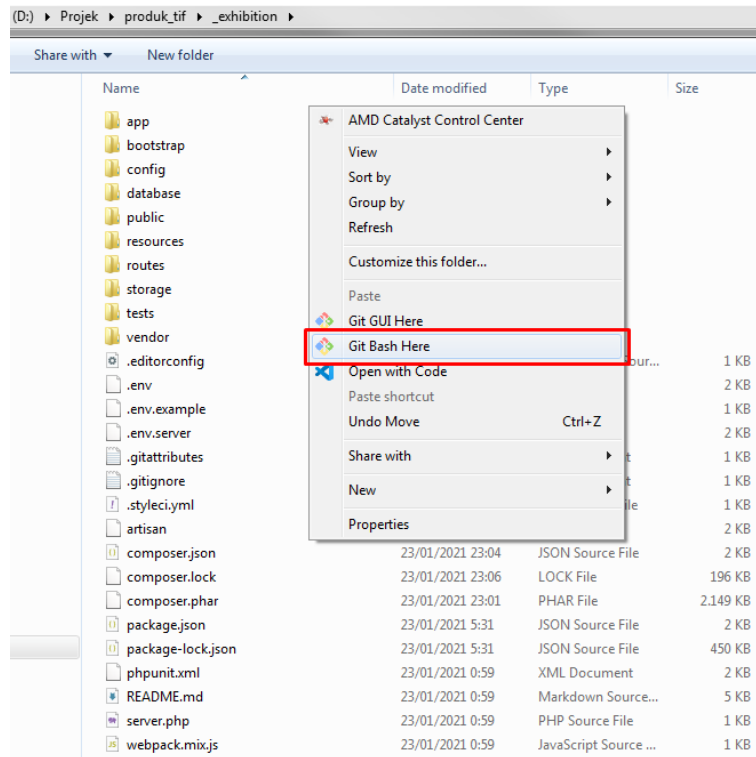
```
git remote add origin https://github.com/labksi/produktif.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

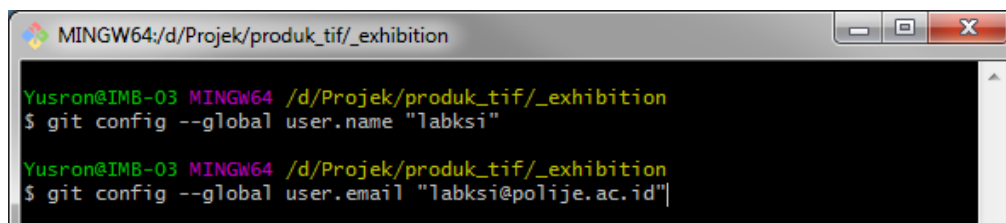
4. Arahkan pada direktori tempat project akan diupload ke repository di Github.
5. Untuk pengguna Windows klik kanan folder project dan pilih Git Bash. Berikut gambarnya:



6. Jika baru pertama kali menggunakan software Git, sebaiknya konfigurasi username dan email dulu :

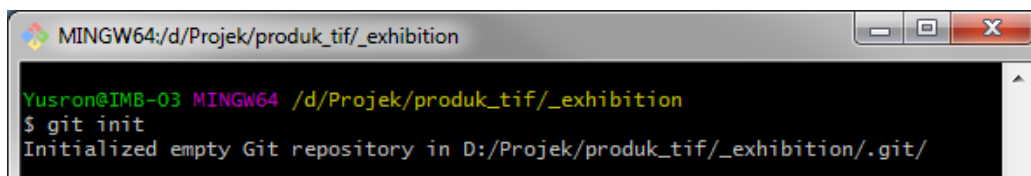
```
git config --global user.name "username anda"
```

```
git config --global user.email "email anda"
```



7. Setelah melakukan konfigurasi username dan email, kemudian lakukan inisiasi dengan menulis perintah berikut

```
git init
```



8. Kemudian tambahkan semua file yang ada dalam folder project kita, ketikan

```
git add *
```



```
MINGW64:/d/Projek/produk_tif/_exhibition

Yusron@IMB-03 MINGW64 /d/Projek/produk_tif/_exhibition (master)
$ git add *
The following paths are ignored by one of your .gitignore files:
vendor
Use -f if you really want to add them.
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in app/Console/Kernel.php.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in app/Exceptions/Handler.php.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in app/Http/Controllers/Auth/ConfirmPasswor
dController.php.
```

9. Kemudian commit projectnya, sebagai contoh disini commit message "First Upload"
- ```
git commit -m "First Upload"
```

```
MINGW64:/d/Projek/produk_tif/_exhibition

Yusron@IMB-03 MINGW64 /d/Projek/produk_tif/_exhibition (master)
$ git commit -m "First Upload"
```

10. Setelah kita buat commit untuk projectnya, remote repository yang telah dibuat tadi
- ```
git remote add origin https://github.com/labksi/produktif.git
```

```
MINGW64:/d/Projek/produk_tif/_exhibition

Yusron@IMB-03 MINGW64 /d/Projek/produk_tif/_exhibition (master)
$ git remote add origin https://github.com/labksi/produktif.git
```

11. Buatlah cabang repository master
- ```
git branch -M master
```

```
MINGW64:/d/Projek/produk_tif/_exhibition

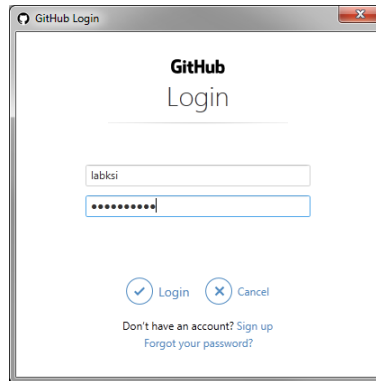
Yusron@IMB-03 MINGW64 /d/Projek/produk_tif/_exhibition (master)
$ git branch -M master
```

12. Setelah meremote repository, kemudian kita pull projectnya dengan perintah
- ```
git pull origin master
```
13. Kirim project ke repository kita.
- ```
git push origin master
```

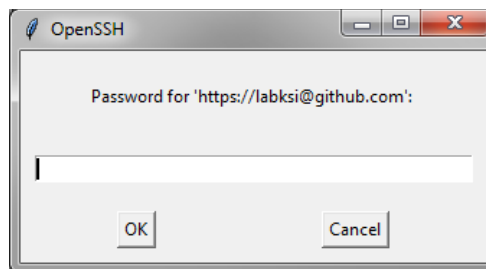
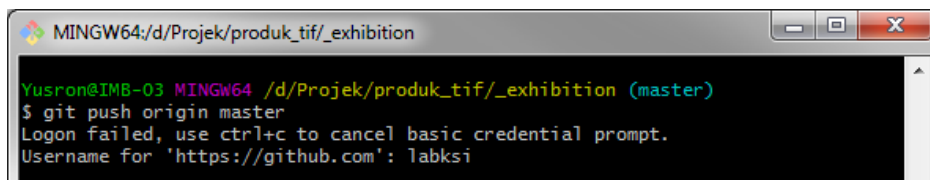
```
MINGW64:/d/Projek/produk_tif/_exhibition

Yusron@IMB-03 MINGW64 /d/Projek/produk_tif/_exhibition (master)
$ git push origin master
```

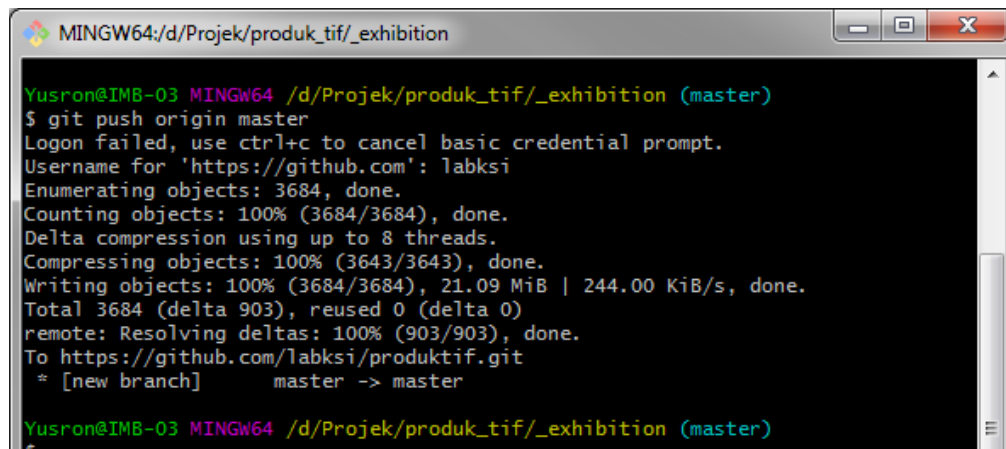
14. Terakhir, kita akan diminta username dan password kita. Untuk password mungkin kita tidak akan melihat password yang kita ketikan



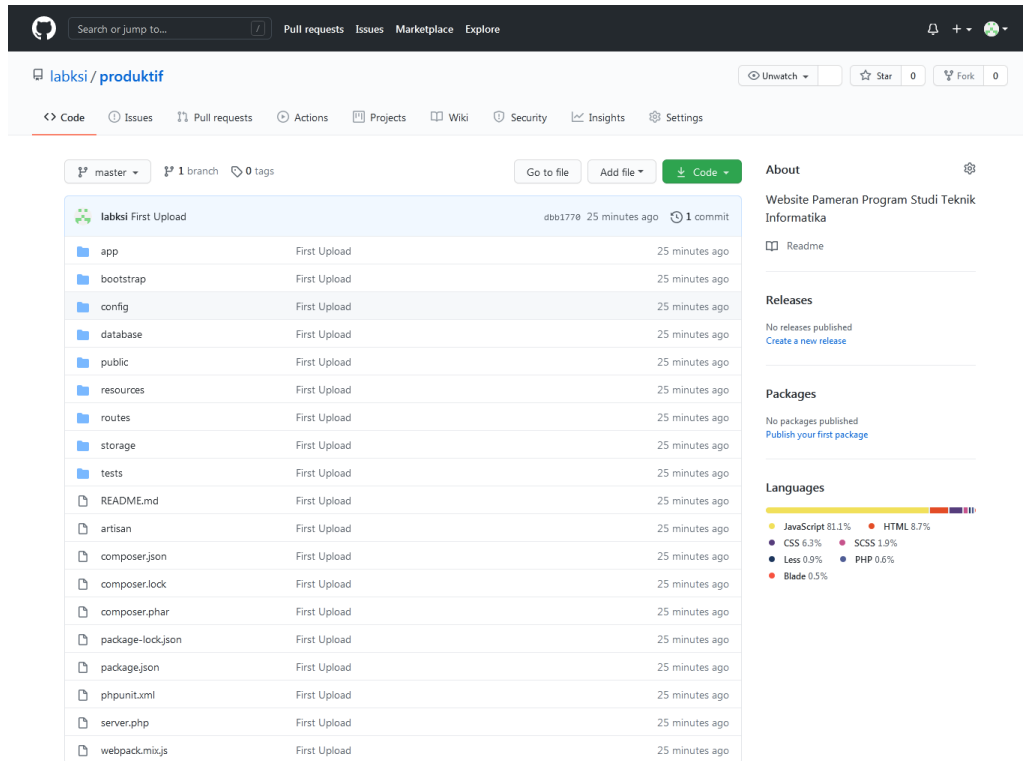
atau jika gagal login tampil seperti berikut :



15. Tunggu sampai project selesai di upload



16. Jika proses upload selesai, cek pada Github lalu refresh repository, maka file kita sudah terupload disana



## g. Hasil dan Pembahasan

### Format Laporan

- Kertas A4
- Format \*.pdf
- Struktur Laporan: Cover, Pendahuluan, Hasil Praktik, Kesimpulan dan Daftar Pustaka
- Berikan Identitas Diri: NIM, Nama, Golongan, Tugas Minggu Ke-X
- Penamaan File: ACARA-X\_GOL\_NIM\_NAMA.pdf  
\*\*Contoh: \*\*ACARA-1\_A\_E1234356\_KHAFID.pdf

Kumpulkan pada <http://jti.poliije.ac.id/elearning/>

## h. Rubrik Penilaian

| Item Penilaian | Kriteria Penilaian | Bobot Penilaian | Penilaian |
|----------------|--------------------|-----------------|-----------|
|----------------|--------------------|-----------------|-----------|



|                        |                                                                                                                                      |      |  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------|------|--|
| Kecakapan Pembelajaran | Mengaplikasikan pengetahuan yang sudah didapatkan dengan mengidentifikasi permasalahan utama dan mempertimbangkan kerumitan yang ada | 30%  |  |
| Literasi               | Sumber referensi dalam mengimplementasikan proyek                                                                                    | 10%  |  |
| Produk                 | Capaian akhir produk sesuai target proyek                                                                                            | 40%  |  |
| Presentasi & Laporan   | Menyajikan informasi yang runtut, lengkap dan jelas dalam bentuk laporan serta penyampain waktu presentasi                           | 20%  |  |
| Total                  |                                                                                                                                      | 100% |  |



## Acara 2

**Materi Pembelajaran : Variabel, Type data dan Operator pada PHP**

**Acara Praktikum/Praktik : 1/2**

**Tempat : Daring/Luring Politeknik Negeri Jember**

**Alokasi Waktu : 2 x 100 menit**

### a. Capaian Pembelajaran Mata Kuliah (CPMK)

- Mahasiswa mampu memahami dan mengimplementasikan PHP dalam pengembangan WEB
- Mahasiswa mampu memahami dan mengimplementasikan Variabel, Type data dan Operator pada PHP

### b. Penilaian Bertahap BNSP (*Skill Passport*)

Nama Skema Sertifikasi: Pengembang Web/Web Developer (SKM-496-028)

| No | Kode Unit       | Nama Unit Kompetensi                      | Elemen Kompetensi                                                                                                                        |
|----|-----------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | J.620100.001.01 | Menganalisis Tools                        | <ul style="list-style-type: none"><li>• Mengidentifikasi tools yang akan digunakan</li><li>• Menggunakan tools perangkat lunak</li></ul> |
| 2  | J.620100.002.01 | Menganalisis Skalabilitas Perangkat Lunak | <ul style="list-style-type: none"><li>• Menggunakan tools perangkat lunak</li><li>• Menganalisis kebutuhan skalabilitas</li></ul>        |

### c. Indikator Penilaian

- Ketepatan dalam memahami dan mengimplementasikan PHP dalam pengembangan WEB
- Ketepatan dalam memahami dan mengimplementasikan Variabel, Type data dan Operator pada PHP

### d. Dasar Teori

PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>. PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Pada awalnya PHP merupakan singkatan dari Personal Home Page. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun



perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: Hypertext Preprocessor. PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. PHP dirilis dalam lisensi PHP License, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang biasa digunakan untuk proyek Open Source. Kemudahan dan kepopuleran PHP sudah menjadi standar bagi programmer web di seluruh dunia. Menurut wikipedia pada februari 2014, sekitar 82% dari web server di dunia menggunakan PHP. PHP juga menjadi dasar dari aplikasi CMS (Content Management System) populer seperti Joomla, Drupal, dan WordPress.

Seperti bahasa pemrograman lainnya, PHP juga memiliki aturan penulisan seperti case sensitivity (perbedaan antara huruf besar dan kecil), cara mengakhiri sebuah baris perintah, serta pengaruh penggunaan spasi dalam membuat kode program PHP. Berikut adalah aturan dasar penulisan kode PHP:

- *Case Sensitivity (perbedaan huruf besar dan kecil) dalam PHP*

PHP membedakan huruf besar dan huruf kecil (case sensitive) untuk penamaan variabel, sehingga \$nama, \$Nama dan \$NAMA akan dianggap sebagai 3 variabel yang berbeda. Sering kali error terjadi karena salah menuliskan nama variabel, yang seharusnya menggunakan huruf kecil ditulis dengan huruf besar. Sering kali error terjadi karena salah menuliskan nama variabel, yang seharusnya menggunakan huruf kecil ditulis dengan huruf besar.

- *Penulisan Baris Perintah dalam PHP*

Statement (baris perintah) di dalam PHP adalah kumpulan perintah PHP yang menginstruksikan PHP untuk melakukan sesuatu. Baris perintah ini bisa terdiri dari satu baris singkat (seperti perintah echo untuk menampilkan text di layar) atau bisa sesuatu yang lebih rumit dan terdiri dari beberapa baris, seperti kondisi if, atau kode perulangan (loop). PHP menggunakan tanda semicolon (titik koma) ";" sebagai tanda akhir baris perintah. Kumpulan baris perintah yang menggunakan tanda kurung kurawal seperti kodisi IF atau perulangan (loop) tidak butuh tanda titik koma setelah kurung penutup.

- *Karakter Spasi dan Tab dalam PHP*





Secara umum, karakter spasi dan tab diabaikan ketika mengeksekusi kode program PHP. Anda boleh memecah sebuah statement menjadi beberapa baris atau menyatukan beberapa statement dalam sebuah baris yang panjang. Kemudian menambahkan beberapa spasi atau tab di awal untuk memudahkan membaca kode program (indenting).

- *Tag Pembuka dan Penutup PHP*

Tag pembuka dalam kode PHP adalah `<?php` dan tag penutup `?>`. Khusus untuk file PHP yang seluruhnya terdiri dari perintah PHP (tidak ada kode HTML di dalamnya), maka tanda kurung penutup `?>` boleh tidak ditulis. Teknik ini sering dipakai untuk menghindari masalah jika sudah sering men-include satu file PHP ke file PHP lain.

## **1. Variabel**

Dalam pemrograman, variabel adalah suatu lokasi penyimpanan (di dalam memori komputer) yang berisikan nilai atau informasi yang nilainya tidak diketahui maupun telah diketahui (wikipedia). Dalam definisi bebasnya, variabel adalah kode program yang digunakan untuk menampung nilai tertentu. Nilai yang disimpan di dalam variabel selanjutnya dapat dipindahkan ke dalam database, atau ditampilkan kembali ke pengguna.

Nilai dari variabel dapat di isi dengan informasi yang diinginkan dan dapat diubah nilainya pada saat kode program sedang berjalan. Sebuah variabel memiliki nama yang digunakan untuk mengakses nilai dari variabel itu. Jika anda memiliki pengetahuan dasar tentang bahasa pemrograman, tentunya tidak asing dengan istilah variabel. Sama seperti variabel dalam bahasa pemrograman lainnya, variabel dalam PHP digunakan untuk menampung nilai inputan dari user, atau nilai yang didefinisikan sendiri. Namun PHP memiliki beberapa aturan tentang cara penggunaan dan penulisan variabel. Beberapa aturan / rule pada penulisan variabel dalam PHP adalah sebagai berikut:

- *Penulisan variabel harus diawali dengan tanda \$*

Variabel di dalam PHP harus diawali dengan dollar sign atau tanda dollar (\$). Setelah tanda \$, sebuah variabel PHP harus diikuti dengan karakter pertama berupa huruf atau underscore (`_`), kemudian untuk karakter kedua dan seterusnya bisa menggunakan huruf, angka atau underscore (`_`). Dengan aturan tersebut, variabel di dalam PHP tidak bisa diawali dengan angka. Minimal panjang variabel adalah 1 karakter setelah tanda \$.

```
<?php
$i;
```



```
$nama;
$Umur;
$_lokasi_memori;
$ANGKA_MAKSIMUM;
?>
```

- *Variabel dalam PHP bersifat case sensitif*

PHP membedakan variabel yang ditulis dengan huruf besar dan kecil (bersifat case sensitif) , sehingga \$belajar tidak sama dengan \$Belajar dan \$BELAJAR, ketiganya akan dianggap sebagai variabel yang berbeda. Untuk menghindari kesalahan program yang dikarenakan salah merujuk variabel, disarankan menggunakan huruf kecil untuk seluruh nama variabel.

```
<?php
$andi="Andi";
echo $Andi; // Notice: Undefined variable: Andi
?>
```

- *Cara Memberikan Nilai kepada Variabel*

Sama seperti sebagian besar bahasa pemrograman lainnya, untuk memberikan nilai kepada sebuah variabel, PHP menggunakan tanda sama dengan (=). Operator `sama dengan` ini dikenal dengan istilah Assignment Operators.

```
<?php
$nama = "andi";
$umur = 17;
$pesan = "Saya sedang belajar PHP di duniaikom.com";
?>
```

Perintah pemberian nilai kepada sebuah variabel disebut dengan assignment. Jika variabel tersebut belum pernah digunakan, dan langsung diberikan nilai awal, maka disebut juga dengan proses inisialisasi.

- *Variabel dalam PHP tidak memerlukan deklarasi terlebih dahulu*

Jika anda pernah mempelajari bahasa pemrograman desktop seperti Pascal, C, C++, dan Visual Basic, di dalam bahasa pemrograman tersebut, sebuah variabel harus dideklarasikan terlebih dahulu sebelum digunakan. Namun di dalam PHP, variabel tidak perlu dideklarasikan terlebih dahulu. Anda bebas membuat variabel baru di tengah-tengah kode program, dan langsung menggunakannya tanpa di deklarasikan terlebih dahulu.



```
<?php
 $andi="Andi";
 echo $andi;
?>
```

- *Variabel dalam PHP tidak bertipe*

Dalam kelompok bahasa pemrograman, PHP termasuk Loosely Type Language, yaitu jenis bahasa pemrograman yang variabelnya tidak terikat pada sebuah tipe tertentu. Hal ini berbeda jika dibandingkan dengan bahasa pemrograman desktop seperti Pascal atau C, dimana jika anda membuat sebuah variabel bertipe integer, maka variabel itu hanya bisa menampung nilai angka, dan anda tidak akan bisa mengisinya dengan huruf.

```
<?php
 $a = 17; // nilai variabel a berisi angka (integer)
 $a = "aku"; // nilai variabel a diubah menjadi kata (string)
 $a = 17.42; // nilai variabel a diubah menjadi desimal (float)
?>
```

- *Variabel Sistem PHP (Predefined Variables)*

Predefined Variables atau terjemahan bebasnya Variabel Sistem PHP, adalah beberapa variabel yang telah di definisikan secara sistem oleh PHP, dan sebaiknya tidak membuat variabel dengan nama yang sama. Beberapa contoh Predefined Variables dalam PHP adalah: \$GLOBALS, \$\_SERVER, \$\_GET, \$\_POST, \$\_FILES, \$\_COOKIE, \$\_SESSION, \$\_REQUEST, \$\_ENV, \$php\_errormsg, \$HTTP\_RAW\_POST\_DATA, \$http\_response\_header, \$argc, \$argv, \$this.

Daftar list Predefined Variables tersebut saya ambil dari manual PHP di <http://www.php.net/reserved.variables>, di dalam manual tersebut juga dijelaskan bahwa mungkin masih terdapat beberapa variabel sistem PHP selain list di atas, hal ini tergantung dengan jenis web server, versi PHP yang digunakan, dan beberapa faktor lainnya. Namun kebanyakan variabel sistem PHP menggunakan tanda \$\_ pada awal nama variabel, namun tidak selalu.

## **2. Tipe Data**

Setelah sebelumnya dipelajari tentang pengertian variabel, dalam beberapa tutorial kedepannya akan mempelajari tipe data PHP. Sebuah variabel atau konstanta merupakan 'tempat' dari data. Di dalam bahasa pemrograman (dan juga PHP), data yang diinput kedalam variabel atau konstanta akan memiliki tipe tertentu. Tipe-tipe ini nantinya



menentukan bagaimana cara memprosesnya. Beberapa tipe data terdengar familiar, seperti tipe data angka, desimal dan text. Namun juga akan menemukan tipe data lain seperti boolean dan array. Untuk tipe data pertama yang akan dibahas adalah tipe data angka bulat, atau disebut dengan tipe data Integer.

Tipe data integer adalah tipe data yang berupa angka bulat seperti: 1, 22, dan -172. Tipe data integer umum digunakan untuk data dengan angka bulat, seperti harga barang, jumlah stock dan jumlah mahasiswa. Jika data yang dimiliki kemungkinan akan mengandung pecahan, maka tipe data yang digunakan adalah float (akan dibahas dalam tutorial berikutnya). Nilai integer dapat bernilai positif (+) maupun negatif (-). Jika tidak diberi tanda, maka diasumsikan nilai tersebut adalah positif. Berikut contoh penulisan bilangan integer dalam PHP:

```
<?php
 $umur=21;
 $harga=15000;
 $rugi=-500000;

 echo $umur; //21
 echo "
";
 echo $harga; //15000
 echo "
";
 echo $rugi; //-500000
?>
```

Jangkauan angka integer bergantung kepada kemampuan komputasi komputer, namun biasanya dimulai dari -2,147,483,648 sampai +2,147,483,647, atau 32bit. Jika terdapat kemungkinan angka yang dihasilkan dari kode program berada diluar jangkauan ini, sebaiknya menggunakan tipe data float. Secara teknis, jangkauan angka integer ini sama dengan jangkauan tipe data LONG pada bahasa C. Namun dikarenakan bahasa C tidak memberikan spesifikasi khusus seberapa besar jangkauan LONG, anda mungkin mendapat hasil yang berbeda.

Selain digunakan untuk menampung angka dengan base 10 (disebut juga angka desimal), tipe data integer digunakan juga untuk menampung angka base 16 (hexadesimal), base 8 (octal), dan base 2 (binary). Untuk mengetahui nilai maksimal tipe data integer pada komputer, PHP menyediakan konstanta PHP\_INT\_MAX.

```
<?php
print PHP_INT_MAX; // 2147483647
?>
```



Tipe data float (disebut juga tipe data floating point, atau real number) adalah tipe data angka yang memiliki bagian desimal di akhir angka, atau memiliki floating point (floating point adalah istilah dalam bahasa Inggris untuk menyebut tanda "titik" yang menandakan bilangan desimal). Contoh angka float adalah seperti: 0,9 atau 3,14.

Tipe data float cocok digunakan untuk variabel yang akan berisi angka pecahan, seperti nilai IPK, hasil pembagian, atau hasil komputasi numerik yang angkanya tidak bisa ditampung oleh data integer. Sama seperti tipe data integer, jangkauan angka float bergantung kepada komputasi prosesor yang digunakan, walaupun pada umumnya berupa angka mulai dari  $1.7 \times 10^{-308}$  sampai dengan  $1.7^{+308}$  dengan 15 digit keakuratan. Programmer tidak perlu khawatir dengan limit angka float ini, karena selain untuk membuat aplikasi matematis tingkat tinggi, tidak akan menggunakan angka float sampai 15 digit.

PHP mendukung 2 cara penulisan tipe data float, yang pertama yaitu penulisan desimal sehari-hari, seperti 0.17 atau 9.47 dan yang kedua berupa penulisan format scientific notation, seperti 0.314E1, atau 12.0E-3. Penggunaan scientific notation digunakan untuk menyederhanakan penulisan, 0.314E1 adalah sama dengan  $0.314 \times 10^1$  atau 3.14, dan 12.0E-3 sama dengan  $12.0 \times 10^{-3}$ , atau 0.012. Sama seperti tipe data integer, variabel dengan tipe data float juga dapat melakukan operasi numerik seperti penambahan, pembagian, perkalian, dan lain-lain. Berikut contoh penulisan bilangan float dalam PHP:

```
<?php
$angka_float1= 0.78;
$angka_float2= 14.99;
$angka_scientific1=0.314E1;
$angka_scientific2=0.3365E-3;

echo $angka_float1; // 0.78
echo "
";
echo $angka_float2; //14.99
echo "
";
echo $angka_scientific1; //3.14
echo "
";
echo $angka_scientific2; //0.0003365
?>
```

Tipe data string adalah tipe data yang berisi text, kalimat, atau kumpulan karakter. Sebagai contoh, "a", "saya sedang belajar PHP" atau "tUT0r1al pHp?!" semuanya adalah string. Tipe data string mungkin adalah tipe data yang paling sering digunakan, dan



memiliki banyak fitur yang disediakan PHP. Karakter yang didukung saat ini adalah 256 karakter ASCII. List karakter ASCII tersebut dapat dilihat di <http://www.ascii-code.com>.

PHP menyediakan 4 cara penulisan tipe data string, yakni Single Quoted, Double Quoted, Heredoc, dan Nowdoc.

- *Penulisan Tipe Data String dengan Single Quoted*

Penulisan tipe data string menggunakan single quoted atau tanda petik satu (karakter `'`) merupakan cara penulisan string yang paling sederhana. Selanjutnya tinggal membuat sebuah kata atau kalimat, dan menambahkan tanda petik satu di awal dan akhir kalimat. Untuk string yang didalamnya juga terdapat tanda petik satu, namun harus mendahuluinya dengan karakter backslash (`\`) agar tidak dianggap sebagai penutup string. Dan jika di dalam string anda ingin menulis tanda backslash, maka harus menulisnya dengan 2 kali (`\\`). Berikut adalah contoh penulisan tipe data string menggunakan metode single quoted:

```
<?php
$string1='Ini adalah string sederhana';

$string2='Ini adalah string
yang bisa memiliki beberapa
baris';

$string3='Dia berkata: "I\'ll be back"';

$string4='Anda telah berhasil menghapus C:\\xampp\\htdocs';

$string5='Kalimat ini tidak akan pindah ke: \n baris baru';

$string6='Variabel juga tidak otomatis ditampilkan $string1 dan $string3';

echo $string1; echo "
";
echo $string2; echo "
";
echo $string3; echo "
";
echo $string4; echo "
";
echo $string5; echo "
";
echo $string6;
?>
```

- *Penulisan Tipe Data String dengan Double Quoted*

Cara kedua dalam penulisan tipe data string dalam PHP adalah dengan menggunakan Double Quoted atau tanda petik dua (karakter `"`). Walaupun seperti tidak ada perbedaan dengan menggunakan single quote, hasil yang di dapat akan sangat berbeda. Dengan double quoted, PHP akan memproses karakter-karakter khusus seperti carriage return (`\n`),



dan karakter tab (\t) dan juga memproses setiap variabel (yang ditandai dengan tanda \$ didepan kata).

Dikarenakan metode double quoted melakukan pemrosesan terlebih dahulu, maka untuk menampilkan karakter khusus seperti tanda petik (karakter `), tanda dollar (karakter \$) dan tanda-tanda khusus lainnya harus menggunakan backslash (karakter \). Berikut adalah tabel karakter khusus untuk double quoted string:

**Tabel 1 Karakter Khusus untuk Double Quoted String**

| Cara Penulisan String | Karakter Yang Ditampilkan                     |
|-----------------------|-----------------------------------------------|
| \"                    | Karakter Tanda petik dua                      |
| \n                    | Karakter Newline                              |
| \r                    | Karakter Carriage return                      |
| \t                    | Karakter Tab                                  |
| \\                    | Karakter Backslash                            |
| \\$                   | Karakter Dollar Sign                          |
| \{                    | Karakter Pembuka Kurung Kurawal               |
| \}                    | Karakter Penutup Kurung Kurawal               |
| \[                    | Karakter Pembuka Kurung Siku                  |
| \]                    | Karakter Penutup Kurung Kurawal               |
| \0 sampai \777        | Karakter ASCII menggunakan nilai oktal        |
| \x0 sampai \xFF       | Karakter ASCII menggunakan nilai heksadesimal |

Sebagai contoh penggunaan double quoted string, saya akan menggunakan contoh yang sama dengan single quoted string, agar dapat dilihat perbedaannya:

```
<?php
$string1="Ini adalah string sederhana";

$string2="Ini adalah string
yang bisa memiliki beberapa
```



```
baris";

$string3="Dia berkata: \"I'll be back\"";

$string4="Anda telah berhasil menghapus C:\\xampp\\htdocs";

$string5="Kalimat ini akan akan pindah ke: \n baris baru";

$string6="Variabel akan otomatis ditampilkan: $string1 dan $string3";

echo $string1; echo "<br \>";
echo $string2; echo "<br \>";
echo $string3; echo "<br \>";
echo $string4; echo "<br \>";
echo $string5; echo "<br \>";
echo $string6;
?>
```

- *Penulisan Tipe Data String Dengan Heredoc*

Cara penulisan tipe data string yang ketiga yaitu dengan fitur PHP yang disebut heredoc. Fitur ini digunakan untuk membuat tipe data string yang dapat berisi beberapa baris kalimat. Dibandingkan dengan menggunakan single quote dan double quote, pembuatan string dengan heredoc tidak terlalu sering digunakan. Agar lebih jelas, berikut adalah contoh penulisan tipe data string dengan heredoc:

```
<?php
$IPK=3.9;
$string1 = <<<end
Saya sedang belajar PHP
di Politeknik Negeri Jember

Kali ini tentang pembahasan
mengenai "PHP",
 dan berharap
bisa dapat IPK $IPK :)
end;

echo $string1;
?>
```

- *Penulisan Tipe Data String Dengan Nowdoc*

Cara penulisan tipe data keempat dalam PHP yaitu dengan fitur Nowdoc. Fitur ini hampir sama dengan Heredoc, namun dengan pengecualian: karakter khusus dan variabel tidak akan diproses oleh PHP, atau mirip dengan single quoted string. Berikut adalah contoh penulisan tipe data string menggunakan metode Nowdoc:





```
<?php
$IPK=3.9;
$string1 = <<< 'selesai'
Saya sedang belajar PHP
di \n Politeknik Negeri Jember

Kali ini tentang pembahasan
mengenai "PHP",
 dan berharap
bisa dapat IPK $IPK :)
selesai;

echo $string1;
?>
```

### 3. Operator

Dalam bahasa pemrograman, terdapat istilah operand dan operator. Operand adalah nilai asal yang digunakan didalam proses operasi, sedangkan operator adalah instruksi yang diberikan untuk mendapatkan hasil dari proses tersebut. Contohnya, operasi: 5+2. Angka 5 dan 2 adalah operand, sedangkan tanda tambah (karakter +) adalah operator. Beberapa operator bisa mengubah nilai dari operandnya sendiri, walaupun kebanyakan hanya sebagai penghubung antar operand. Operator di dalam PHP banyak meminjam contoh karakter dari bahasa C dan Perl. Berdasarkan jumlah operand, operator dapat dibedakan menjadi 3, yaitu Operator Unary, Binary dan Ternary:

- Operator unary adalah operator yang hanya memiliki 1 operand, contohnya karakter – (tanda minus). Tanda minus digunakan membuat sebuah angka menjadi negatif, contohnya: -5, atau karakter + untuk menegaskan nilai positif, contohnya: +5.
- Operator binary adalah operator yang memiliki 2 operand. Operator jenis ini adalah yang paling banyak digunakan, misalkan 5×2, atau 10/3.
- Operator Ternary adalah operator yang memiliki 3 operand. Didalam PHP hanya dikenal 1 operator ternary, yaitu operator kondisi (? :).

| Operator | Name           | Example    | Result                                    |
|----------|----------------|------------|-------------------------------------------|
| +        | Addition       | \$x + \$y  | Sum of \$x and \$y                        |
| -        | Subtraction    | \$x - \$y  | Difference of \$x and \$y                 |
| *        | Multiplication | \$x * \$y  | Product of \$x and \$y                    |
| /        | Division       | \$x / \$y  | Quotient of \$x and \$y                   |
| %        | Modulus        | \$x % \$y  | Remainder of \$x divided by \$y           |
| **       | Exponentiation | \$x ** \$y | Result of raising \$x to the \$y'th power |



Operator Aritmatika adalah operator matematis yang terdiri dari operator penambahan, pengurangan, perkalian, pembagian, modulus, dan eksponensial. Kebanyakan operator aritmatika dalam PHP bertipe binary yakni membutuhkan 2 operand, kecuali operator minus (-) dan plus (+) yang merupakan operator tipe unary (hanya membutuhkan 1 operand). Dari ke 6 operator aritmatika dalam PHP tersebut, operator modulus (\$a % \$b) mungkin terdengar baru. Operator ini menghasilkan sisa hasil bagi dari hasil pembagian. Misalkan 10 % 3, hasilnya adalah 1. Biasanya operator modulus ini digunakan bersama-sama dengan operator pembagian (/).

```
<?php
$hasil1= -3;
$hasil2=3+5;
$hasil3=8-4.5;
$hasil4=2*5;
$hasil5=3+8/5-3;
$hasil6=10 % 4;

echo "\$hasil1: "; var_dump($hasil1); // $hasil1:int(-3)
echo "<br \>";
echo "\$hasil2: "; var_dump($hasil2); // $hasil2:int(8)
echo "<br \>";
echo "\$hasil3: "; var_dump($hasil3); // $hasil3:float(3.5)
echo "<br \>";
echo "\$hasil4: "; var_dump($hasil4); // $hasil4:int(10)
echo "<br \>";
echo "\$hasil5: "; var_dump($hasil5); // $hasil5:float(1.6)
echo "<br \>";
echo "\$hasil6: "; var_dump($hasil6); // $hasil6:int(2)
?>
```

| Assignment          | Same as...             | Description                                                           |
|---------------------|------------------------|-----------------------------------------------------------------------|
| <code>x = y</code>  | <code>x = y</code>     | The left operand gets set to the value of the expression on the right |
| <code>x += y</code> | <code>x = x + y</code> | Addition                                                              |
| <code>x -= y</code> | <code>x = x - y</code> | Subtraction                                                           |
| <code>x *= y</code> | <code>x = x * y</code> | Multiplication                                                        |
| <code>x /= y</code> | <code>x = x / y</code> | Division                                                              |
| <code>x %= y</code> | <code>x = x % y</code> | Modulus                                                               |



Operator gabungan assignment adalah cara penulisan singkat operator dengan menggunakan sebuah operator assignment secara bersamaan dengan operator lainnya. Dalam PHP, operator gabungan ini adalah antara operator assignment dengan operator lain seperti operator aritmatika, string, bitwise, dll. Untuk memahami operator gabungan assignment ini, perhatikan contoh berikut:

```
<?php
$a = 10;
$a = $a + 5;
$a = $a - 10;
echo "\$a = $a";
echo "
";

// sama hasilnya dengan kode berikut:

$b = 10;
$b += 5; // sama dengan $b = $b + 5;
$b -= 10; // sama dengan $b = $b - 10;
echo "\$b = $b";
?>
```

Pada baris ke... digunakan operator gabungan untuk mempersingkat penulisan `$b = $b + 5` menjadi `$b += 5;`. Operator gabungan assignment bisa digunakan hampir untuk seluruh operator lainnya. Operator gabungan ini akan menghemat penulisan kode program, namun jika andanyaman menggunakan kode yang sedikit panjang juga tidak masalah.



| Operator | Name                     | Example     | Result                                                                                                                                              |
|----------|--------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| ==       | Equal                    | \$x == \$y  | Returns true if \$x is equal to \$y                                                                                                                 |
| ===      | Identical                | \$x === \$y | Returns true if \$x is equal to \$y, and they are of the same type                                                                                  |
| !=       | Not equal                | \$x != \$y  | Returns true if \$x is not equal to \$y                                                                                                             |
| <>       | Not equal                | \$x <> \$y  | Returns true if \$x is not equal to \$y                                                                                                             |
| !==      | Not identical            | \$x !== \$y | Returns true if \$x is not equal to \$y, or they are not of the same type                                                                           |
| >        | Greater than             | \$x > \$y   | Returns true if \$x is greater than \$y                                                                                                             |
| <        | Less than                | \$x < \$y   | Returns true if \$x is less than \$y                                                                                                                |
| >=       | Greater than or equal to | \$x >= \$y  | Returns true if \$x is greater than or equal to \$y                                                                                                 |
| <=       | Less than or equal to    | \$x <= \$y  | Returns true if \$x is less than or equal to \$y                                                                                                    |
| <=>      | Spaceship                | \$x <=> \$y | Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7. |

Operator perbandingan membandingkan nilai dari 2 operand. Hasilnya selalu salah satu dari TRUE atau FALSE. Hasil perbandingan akan bernilai TRUE jika kondisi perbandingan tersebut benar, atau FALSE jika kondisinya salah. Operand untuk operator perbandingan ini bisa berupa tipe data angka (integer atau float), maupun bertipe string. Operator perbandingan akan memeriksa nilai dan (untuk beberapa operator) juga tipe data dari operand. Operator perbandingan ini sering digunakan untuk kondisi IF, misalkan jika \$nama\_user ada di dalam database, maka jalankan kode program. Berikut adalah beberapa contoh penggunaan operator perbandingan dalam PHP:

```
<?php
echo "1. 12 < 14 = "; var_dump(12<14); // bool(true)
echo "
";

echo "2. 14 < 14 = "; var_dump(14<14); // bool(false)
echo "
";

echo "3. 14 <= 14 = "; var_dump(14<=14); // bool(true)
```



```
echo "
";

echo "4. 10 <> '10' = "; var_dump(10<>'10'); // bool(false)
echo "
";

echo "5. 10 == '10' = "; var_dump(10=='10'); // bool(true)
echo "
";

echo "6. 10 === '10' = "; var_dump(10==='10'); // bool(false)
echo "
";

echo "7. '150' == '1.5e2' = "; var_dump('150'=='1.5e2'); // bool(true)
echo "
";

echo "8. 'duniaikom' == 0 = "; var_dump('duniaikom'==0); // bool(true)
echo "
";
?>
```

Operator Increment dan Decrement adalah penyebutan untuk operasi seperti  $a++$ , dan  $a--$ . Jika anda telah mempelajari bahasa pemrograman lain, operasi increment dan decrement ini sering digunakan dalam perulangan (looping). Dari tabel diatas terlihat bahwa terdapat 2 jenis increment, yaitu Pre-increment, dan Post-Increment, dan 2 jenis decrement, yaitu Pre-decrement dan Post-decrement. Perbedaan keduanya terletak pada posisi mana tanda tambah atau kurang diletakkan.

| Operator | Name           | Description                                  |
|----------|----------------|----------------------------------------------|
| $++\$x$  | Pre-increment  | Increments $\$x$ by one, then returns $\$x$  |
| $\$x++$  | Post-increment | Returns $\$x$ , then increments $\$x$ by one |
| $--\$x$  | Pre-decrement  | Decrements $\$x$ by one, then returns $\$x$  |
| $\$x--$  | Post-decrement | Returns $\$x$ , then decrements $\$x$ by one |

Increment digunakan untuk menambah variabel sebanyak 1 angka, sedangkan decrement digunakan untuk mengurangi variabel sebanyak 1 angka. Penulisannya menggunakan tanda tambah 2 kali untuk increment, dan tanda kurang 2 kali untuk decrement. Penempatan tanda tambah atau kurang ini boleh diawal, atau diakhir variabel, namun keduanya memiliki perbedaan, sehingga terdapat 4 jenis increment dan decrement dalam PHP. Untuk memahami cara penggunaan operator increment dan decrement, berikut contoh kode program PHP:



```
<?php
echo "<h3>Postincrement</h3>";
$a = 5;
echo "\$a = $a
";
echo "\$a akan bernilai 5: " . $a++ . " (\$a++)
";
echo "\$a akan bernilai 6: " . $a . "
";

echo "<h3>Preincrement</h3>";
$a = 5;
echo "\$a = $a
";
echo "\$a akan bernilai 6: " . ++$a . " (++\$a)
";
echo "\$a akan bernilai 6: " . $a . "
";

echo "<h3>Postdecrement</h3>";
$a = 5;
echo "\$a = $a
";
echo "\$a akan bernilai 5: " . $a-- . " (\$a--)
";
echo "\$a akan bernilai 4: " . $a . "
";

echo "<h3>Predecrement</h3>";
$a = 5;
echo "\$a = $a
";
echo "\$a akan bernilai 4: " . --$a . " (--\$a)
";
echo "\$a akan bernilai 4: " . $a . "
";
?>
```

| Operator | Name | Example     | Result                                          |
|----------|------|-------------|-------------------------------------------------|
| and      | And  | \$x and \$y | True if both \$x and \$y are true               |
| or       | Or   | \$x or \$y  | True if either \$x or \$y is true               |
| xor      | Xor  | \$x xor \$y | True if either \$x or \$y is true, but not both |
| &&       | And  | \$x && \$y  | True if both \$x and \$y are true               |
|          | Or   | \$x    \$y  | True if either \$x or \$y is true               |
| !        | Not  | !\$x        | True if \$x is not true                         |

Operator logika digunakan untuk membandingkan 2 inputan kondisi boolean, yakni true atau false. Dalam bagian ini akan dibahas tentang jenis dan cara penggunaan operator logika (Logical Operators) dalam PHP, serta memahami pengertian prinsip short-circuit



dalam operasi logika PHP. Jadi operator Logika adalah operator yang digunakan untuk membandingkan 2 kondisi logika, yaitu logika benar (TRUE) dan logika salah (FALSE).

Operator logika sering digunakan untuk kondisi IF, atau untuk keluar dari proses perulangan (looping). Jenis operand dalam operator logika ini adalah variabel dengan tipe boolean. Namun jika operand bukan boolean, akan "dikonversi" menjadi boolean oleh PHP. Perbedaan dari operator AND dengan &&, dan OR dengan || terkait dengan cara penulisan dan aturan "kekuatan" operator. Operator && dan || memiliki "kekuatan" lebih tinggi dari pada AND dan OR, sehingga baris perintah: \$a AND \$b || \$c, akan dieksekusi oleh PHP menjadi \$a AND (\$b || \$c). Berikut adalah contoh kode program, cara penggunaan operator logika dalam PHP:

```
<?php
$hasil1 = true and false;
echo '$hasil1 = ';
echo var_dump($hasil1)."
"; // $hasil1 = bool(true)

$hasil2 = (true and false);
echo '$hasil2 = ';
echo var_dump($hasil2)."
"; // $hasil2 = bool(false)

$hasil3 = (true xor false);
echo '$hasil3 = ';
echo var_dump($hasil3)."
"; // $hasil3 = bool(true)

$hasil4 = (false or true && false);
echo '$hasil4 = ';
echo var_dump($hasil4)."
"; // $hasil4 = bool(false)

$a=true;
$b=false;
$hasil5 = ($a and $b || $a or b);
echo '$hasil5 = ';
echo var_dump($hasil5); // $hasil5 = bool(true)
?>
```

String merupakan tipe dasar yang paling sering digunakan dalam PHP. Dalam PHP, hanya terdapat 1 jenis operator String, yakni operasi penyambungan (concatenation) string. Operator ini menggunakan karakter titik (.) atau (.=). Operator penyambungan string ini membutuhkan 2 inputan yang bertipe data string. Hasil dari operator ini adalah sebuah string yang terdiri dari sambungan kedua string tersebut. Berikut adalah contoh kode program cara penggunaan operator string dalam PHP:



| Operator | Name                     | Example          | Result                             |
|----------|--------------------------|------------------|------------------------------------|
| .        | Concatenation            | \$txt1 . \$txt2  | Concatenation of \$txt1 and \$txt2 |
| .=       | Concatenation assignment | \$txt1 .= \$txt2 | Appends \$txt2 to \$txt1           |

```
<?php
$a = "Hello ";
$b = "World!"
$a .= $b;
echo $a; // Hello World!
echo "
";

$a = "belajar ";
$b = "PHP ";
$c = "di Politeknik Negeri Jember";
$hasil= "Saya sedang ".$a.$b.$c;
echo $hasil; // Saya sedang belajar PHP di Politeknik Negeri Jember
?>
```

Cara Alternatif penyambung string adalah dengan kurung kurawal { }. Didalam PHP, tanda kurung kurawal (karakter { dan }) untuk variabel bisa berfungsi sebagai penyambung string. Contoh kode program diatas dapat juga ditulis menjadi:

```
<?php
$a = "Hello ";
$hasil = "{$a} World!";
echo $hasil; // Hello World!
echo "
";

$a = "belajar ";
$b = "PHP ";
$c = "di Politeknik Negeri Jember";
$hasil= "Saya sedang {$a}{$b}{$c}";
echo $hasil; // Saya sedang belajar PHP di Politeknik Negeri Jember
?>
```

Beberapa situs referensi PHP, menyebutkan bahwa cara ini “lebih cepat” untuk menyambung string daripada menggunakan operator titik (.), Namun perbedaannya tidak akan terlihat untuk kode program sederhana.

#### e. Alat dan Bahan

- Client Server: xampp





- Text Editor: Visual Code/Sublime Text 3
- Management Storage: Git dan Github
- Tools Dependency Manager Multiplatform: Composer
- Kertas A4 / Folio Bergaris
- Pulpen

#### **f. Prosedur Kerja**

##### **Nama File : variabel.php**

```
<?php
$nim="E12345678";
$nama="Budi";
$prodi="Teknik Informatika";
echo "Selamat Praktikum";
echo $nim;
echo $nama;
echo $prodi;
?>
```

##### **Nama File : typedata.php**

```
<?php
$x = 5985;
var_dump($x);
echo "
";
$x = -345; // negative number
var_dump($x);
echo "
";
$x = 0x8C; // hexadecimal number
var_dump($x);
echo "
";
$x = 047; // octal number
var_dump($x);
$x = 10.365;
var_dump($x);
echo "
";
$x = 2.4e3;
var_dump($x);
echo "
";
$x = 8E-5;
var_dump($x);
echo strlen("Hello world!");
echo strpos("Hello world!","world");
?>
```



**Nama File : operator1.php**

```
<?php
$tugas1=90;
$tugas2=80;
$jumlah=$tugas1 + $tugas2;
$rerata=$jumlah/2 ;
echo "Nilai Tugas I :".$tugas1."
";
echo "Nilai Tugas II: ".$tugas2."
";
echo "Jumlah Tugas :".$jumlah."
";
echo "Rerata Tugas :".$rerata;
?>
```

**Nama File : operator2.php**

```
<?php
$x=10;
$y=6;
echo ($x + $y);
echo ($x - $y);
echo ($x * $y);
echo ($x / $y);
echo ($x % $y);
$a = "Hello";
$b = $a . " world!";
echo $b;
Echo $a. " -- " . $b. " ini string operator ";
?>
```

**Tugas**

1. Tanda " " pada nilai variabel biasanya menandakan sebuah string, tetapi pada saat dilakukan operasi penjumlahan ternyata menghasilkan nilai jumlah seperti pada umumnya penjumlahan aritmatika, mengapa bisa terjadi? (variable dan tipe data)
2. Apa yang membedakan pembagian dengan operator / dan operator modulo % ? (variable dan tipe data)
3. Tambahkan kode program operator agar bisa menghasilkan nilai gabungan string tugas1 dan tugas2 menjadi 9080? (operator)



#### g. Hasil dan Pembahasan

##### Format Laporan

Kertas A4

Format \*.pdf

Struktur Laporan: Cover, Pendahuluan, Hasil Praktik, Kesimpulan dan Daftar Pustaka

Berikan Identitas Diri: NIM, Nama, Golongan, Tugas Minggu Ke-X

Penamaan File: ACARA-X\_GOL\_NIM\_NAMA.pdf

**\*\*Contoh: \*\***ACARA-1\_A\_E1234356\_KHAFID.pdf

Kumpulkan pada <http://jti.poliije.ac.id/elearning/>

#### h. Rubrik Penilaian

| Item Penilaian         | Kriteria Penilaian                                                                                                                   | Bobot Penilaian | Penilaian |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------|
| Kecakapan Pembelajaran | Mengaplikasikan pengetahuan yang sudah didapatkan dengan mengidentifikasi permasalahan utama dan mempertimbangkan kerumitan yang ada | 30%             |           |
| Literasi               | Sumber referensi dalam mengimplementasikan proyek                                                                                    | 10%             |           |
| Produk                 | Capaian akhir produk sesuai target proyek                                                                                            | 40%             |           |
| Presentasi & Laporan   | Menyajikan informasi yang runtut, lengkap dan jelas dalam bentuk laporan serta penyampain waktu presentasi                           | 20%             |           |
| Total                  |                                                                                                                                      | 100%            |           |